

Lab Assignment



Cybersecurity Professional Program
Introduction to Python for
Security

Network Communication

PY-06-L1

Server Socket

Copyright © 1996–2021 HackerU Ltd.
All Rights Reserved.

Lab Objective

Learn how to create a socket with Python and wait for a connection from a client.

Lab Mission

Run the required commands to create a listening server.

Lab Duration

15–25 minutes

Requirements

- Basic knowledge of the socket module

Resources

- Environment & Tools
 - Windows, macOS, Linux
 - PyCharm
 - Python 3
 - Netcat
 - ***Kali-linux-2019.3-amd64.iso***



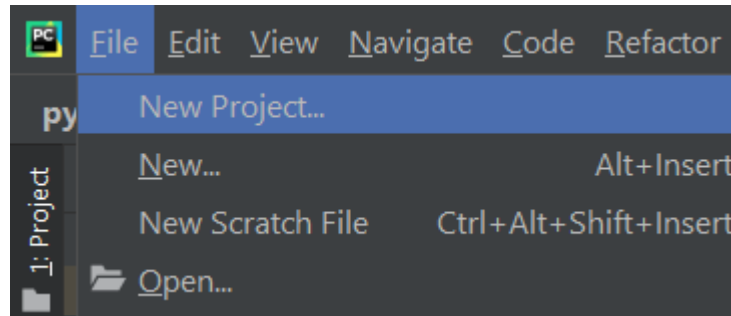
Textbook References

- Chapter 6: Network Communication
 - Section 1: Creating a Server Socket

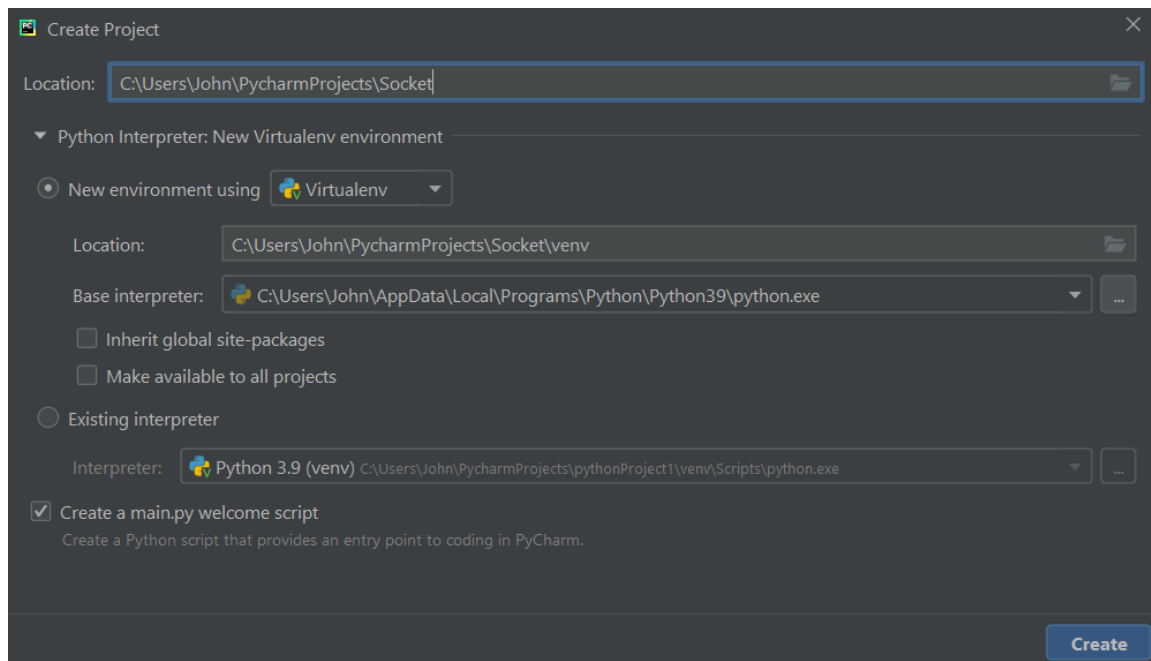
Lab Task 1: Server Socket Creation

In this task, you will define a socket in a server using Python and wait for a connection.

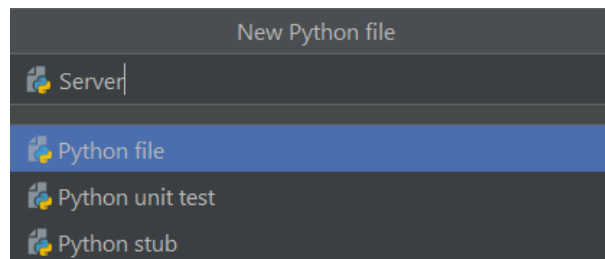
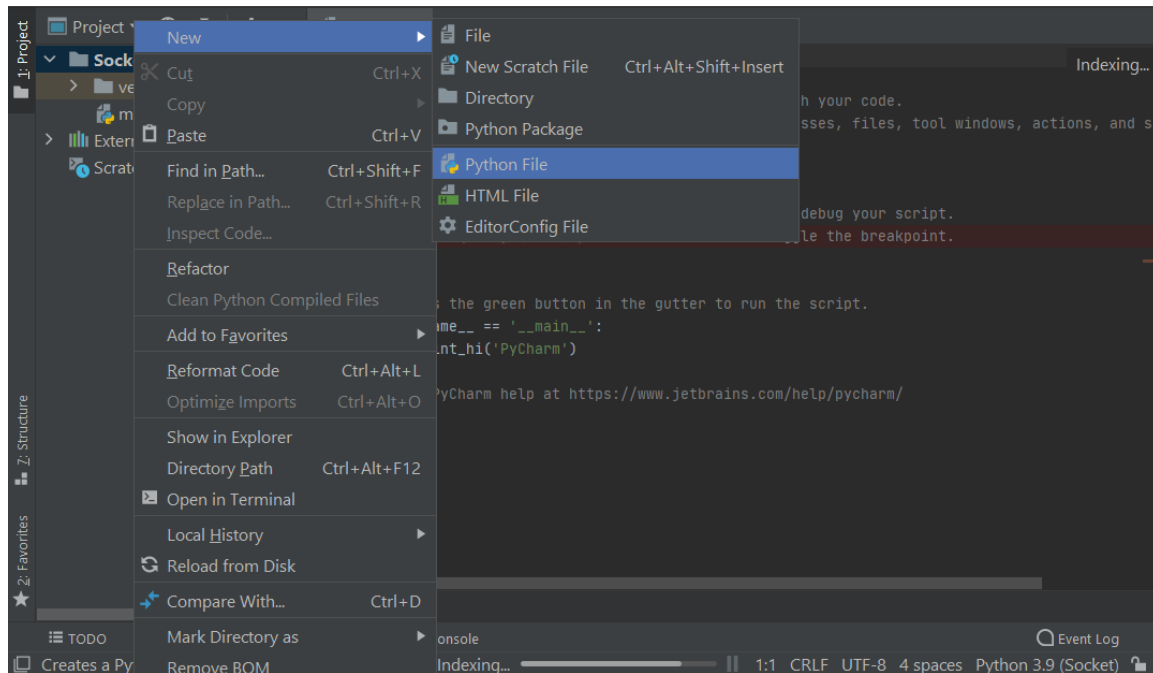
- 1 Open PyCharm and create a new project by clicking **File** and selecting **New Project**.



- 2 Name the project, select **Inherit global site-packages** and **Make available to all projects**, and click **Create**.



- 3 Create a new Python file by right clicking the project's folder and selecting **New** → **Python File**. Name the file **Server**.



- 4 Import the module **socket** to the file.

```
import socket
```

- 5 Create a socket variable.

```
import socket
```

```
my_sock = socket.socket()
```

- 6** Bind the socket to accept connection from all IP addresses on port 4444.

```
import socket  
  
my_sock = socket.socket()  
  
my_sock.bind(("0.0.0.0", 4444))
```

- 7** Allow only one connection to the socket.

```
import socket  
  
my_sock = socket.socket()  
  
my_sock.bind(("0.0.0.0", 4444))  
  
my_sock.listen(1)
```

- 8** Allow a connection to the socket and save the connection object and the address to variables.

```
import socket  
  
my_sock = socket.socket()  
  
my_sock.bind(("0.0.0.0", 4444))  
  
my_sock.listen(1)  
  
connection, address = my_sock.accept()
```

9 Close the connection.

```
import socket

my_sock = socket.socket()

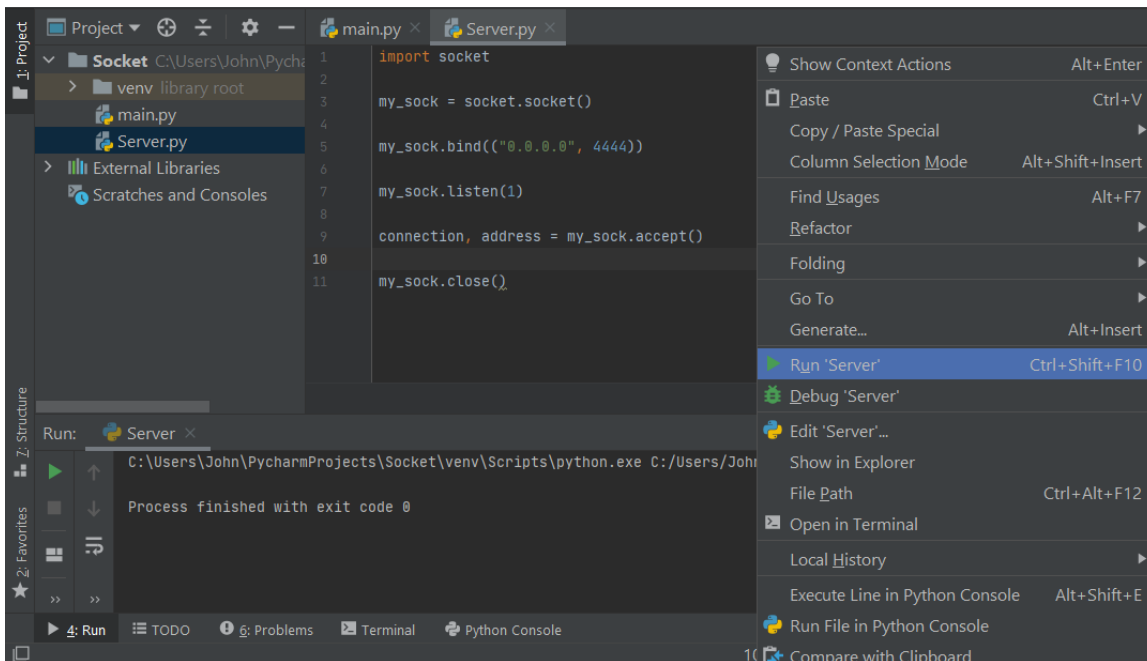
my_sock.bind(("0.0.0.0", 4444))

my_sock.listen(1)

connection, address = my_sock.accept()

my_sock.close()
```

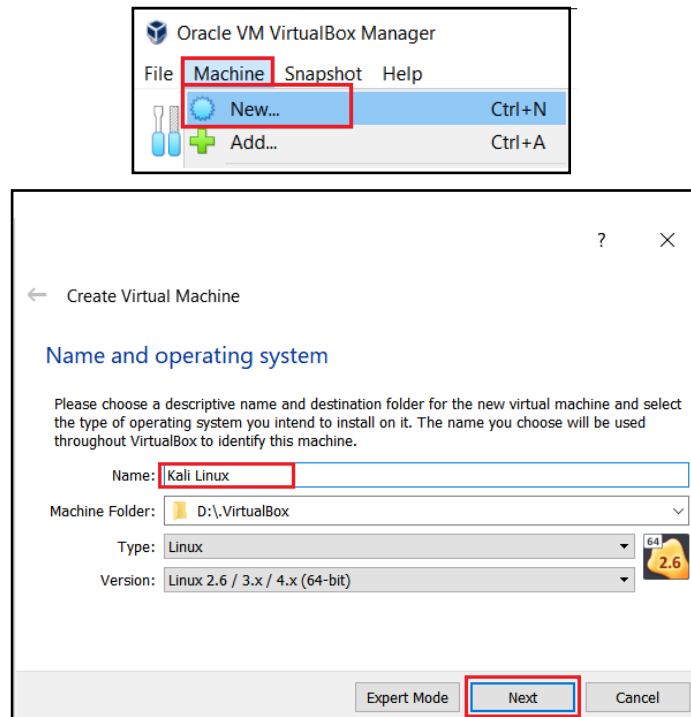
10 Right click the file and run the code.



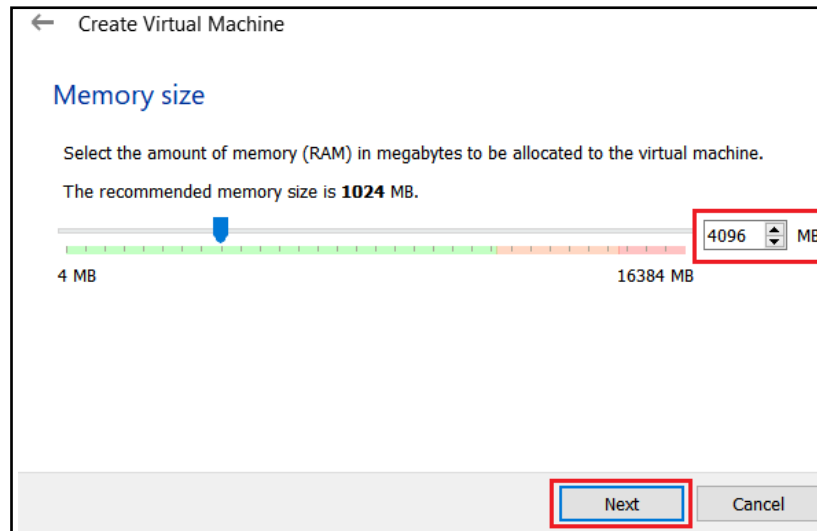
Lab Task 2: Server Socket Creation

In this task, you will define a socket in a server using Python and wait for a connection.

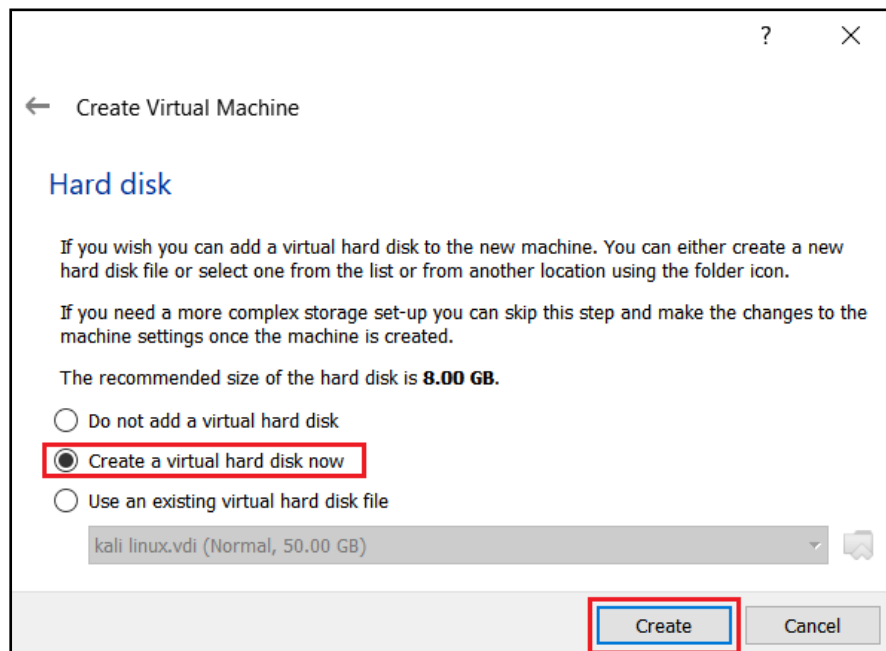
- 1 Open VirtualBox, click the **Machine** tab, click **New**, and name the VM **Kali Linux**.



- 2 Set the memory to **2048 MB** for proper functionality. You can use a larger setting (in accordance with the computer's available resources) for enhanced functionality.



- 3 In the next window, select **Create a virtual hard disk now** and click **Create**.



4 Select **VHD** for the hard disk file type and click **Next**.

← Create Virtual Hard Disk

Hard disk file type

Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

☐ VDI (VirtualBox Disk Image)

☒ VHD (Virtual Hard Disk)

☐ VMDK (Virtual Machine Disk)

Expert Mode **Next** Cancel

5 Select **Dynamically allocated** and click **Next**.

← Create Virtual Hard Disk

Storage on physical hard disk

Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

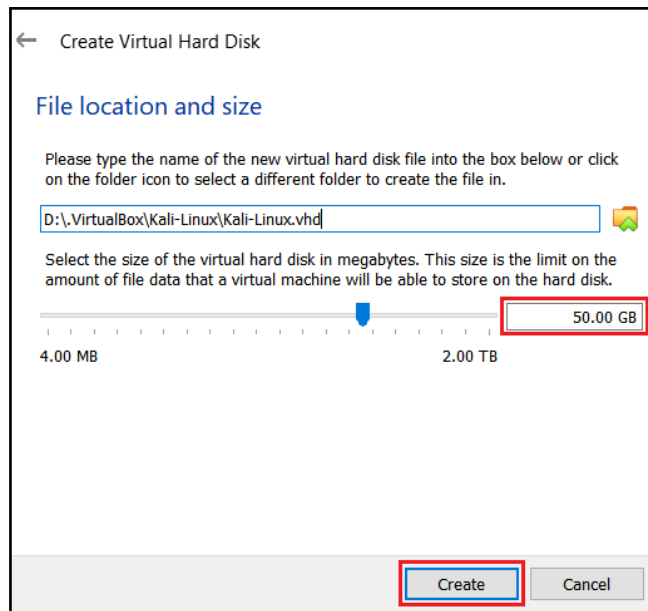
A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

☒ Dynamically allocated

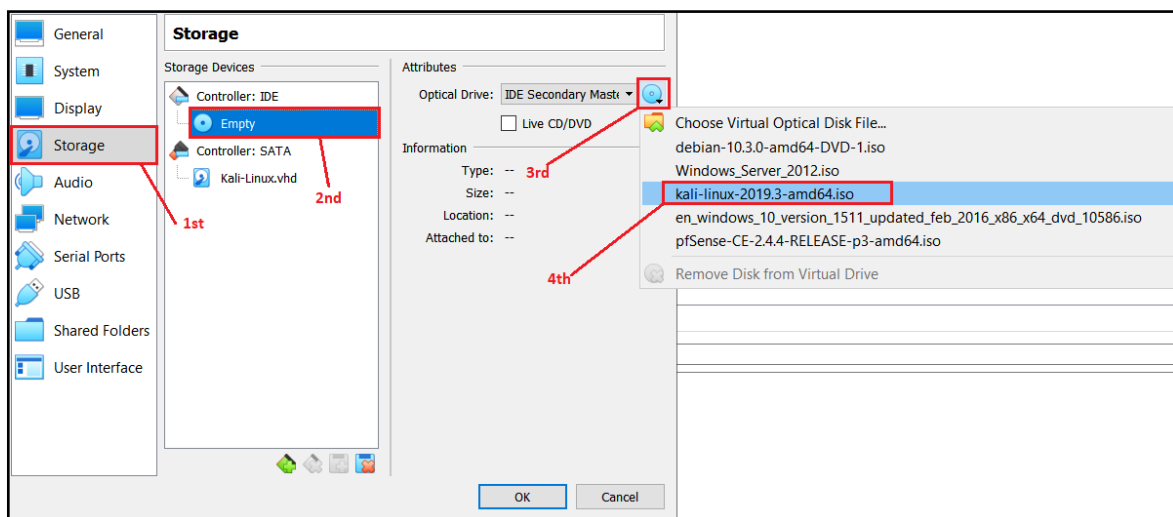
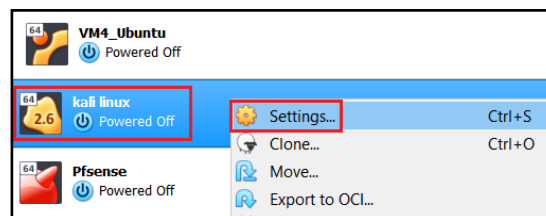
☐ Fixed size

Next Cancel

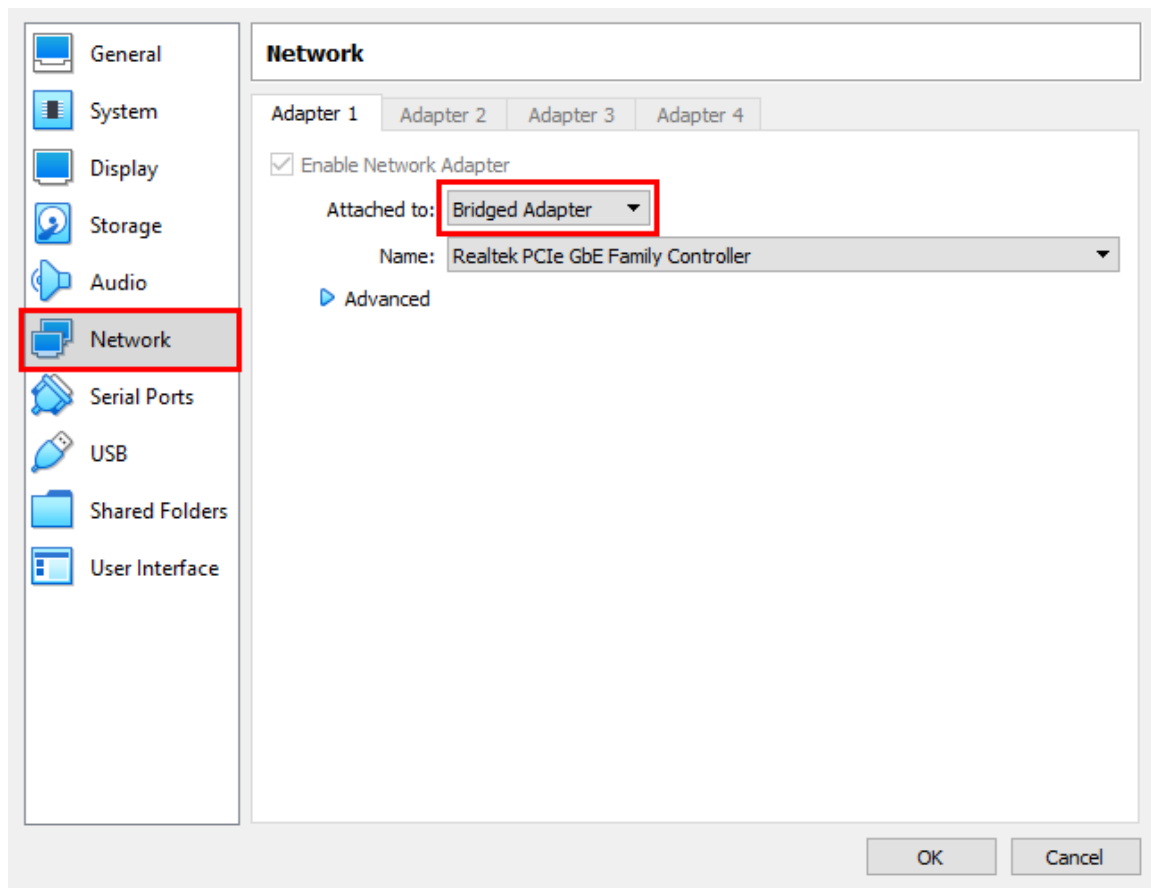
- 6 Select the file location and set the storage size to **50 GB**. If your computer does not have enough storage, you can set it to **20 GB**.



- 7 Right-click the **Kali VM**, go to **Settings**, and insert the **Kali-linux-2019.3-amd64.iso** in the drive.



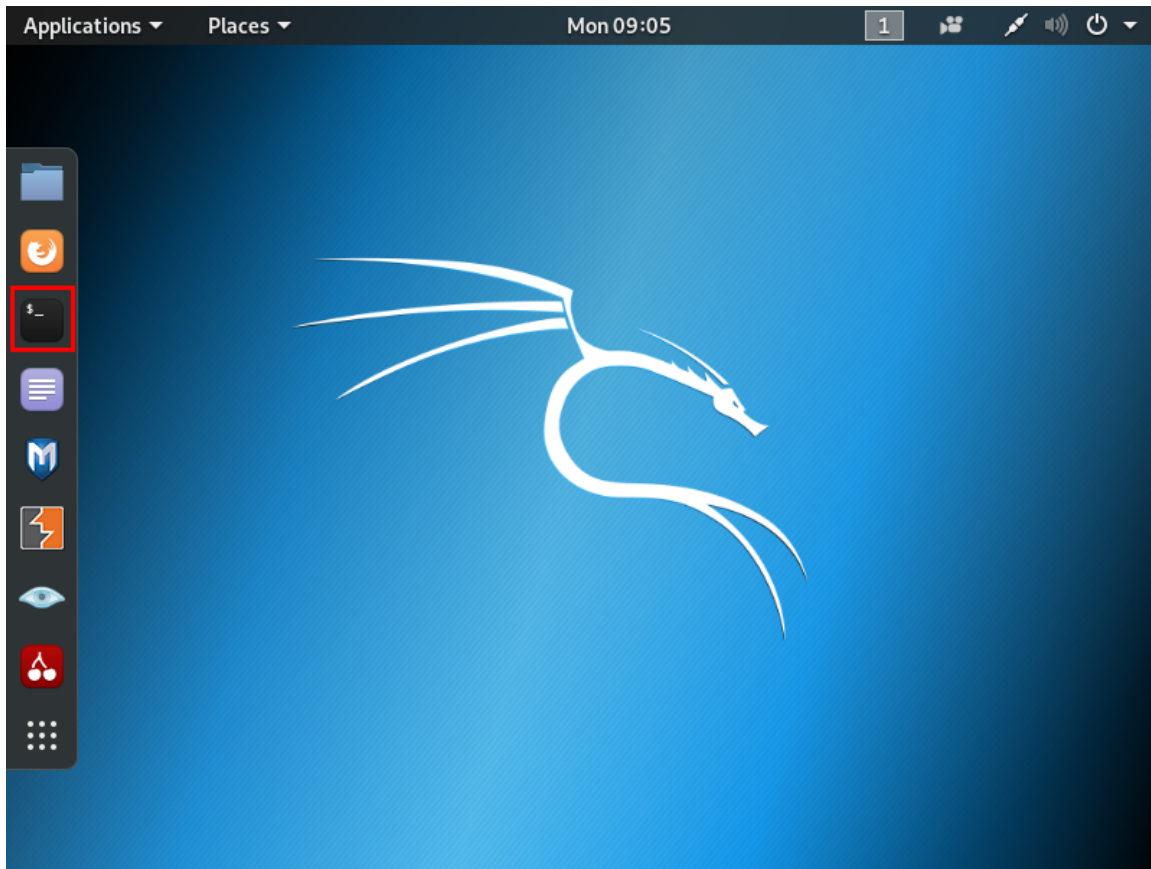
- 8 Click **Network** and make sure the machine is set to **Bridged Adapter**.



- 9 Run the virtual machine and select the live version.



- 10** Open the terminal by clicking its icon in the option bar on left side of the window.



- 11** Run the command ***netcat -v <IP> <port>***.

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# netcat -v 10.0.0.10 4444  
10.0.0.10: inverse host lookup failed: Unknown host  
(UNKNOWN) [10.0.0.10] 4444 (?) open  
root@kali:~#
```